

Лабораторная работа №14

Операционные системы

Мишина А. А.

11 мая 2023

- Мишина Анастасия Алексеевна
- Группа НПИбд-02-22

- Приобретение практических навыков работы с именованными каналами

- Изучите приведённые в тексте программы `server.c` и `client.c`. Взяв данные примеры за образец, напишите аналогичные программы, внося следующие изменения:
 1. Работает не 1 клиент, а несколько (например, два).
 2. Клиенты передают текущее время с некоторой периодичностью (например, раз в пять секунд). Используйте функцию `sleep()` для приостановки работы клиента.
 3. Сервер работает не бесконечно, а прекращает работу через некоторое время (например, 30 сек). Используйте функцию `clock()` для определения времени работы сервера. Что будет в случае, если сервер завершит работу, не закрыв канал?

Выполнение лабораторной работы

```
[aamishina@fedora lab14]$ touch common.h client.c client2.c server.c
```

Рис. 1: Создание файлов common.h, server.c, client.c, client2.c.

```
/*
 * common.h - заголовочный файл со стандартными определениями
 */

#ifndef __COMMON_H__
#define __COMMON_H__

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <errno.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>

#define FIFO_NAME "/tmp/fifo"
#define MAX_BUFF 80

#endif /* __COMMON_H__ */
~
```

Рис. 2: Содержимое заголовочного файла.

```
#include "common.h"

int main()
{
    int readfd; /* дескриптор для чтения из FIFO */
    int n;
    char buff[MAX_BUFF]; /* буфер для чтения данных из FIFO */

    /* баннер */
    printf("FIFO Server...\n");

    /* создаем файл FIFO с открытыми для всех
     * правами доступа на чтение и запись
     */
    if(mknod(FIFO_NAME, S_IFIFO | 0666, 0) < 0)
    {
        fprintf(stderr, "%s: Невозможно создать FIFO (%s)\n", __FILE__, strerror(errno));
        exit(-1);
    }

    /* откроем FIFO на чтение */
    if((readfd = open(FIFO_NAME, O_RDONLY)) < 0)
    {
        fprintf(stderr, "%s: Невозможно открыть FIFO (%s)\n", __FILE__, strerror(errno));
        exit(-2);
    }

    /* читаем данные из FIFO и выводим на экран */
    clock_t now=time(NULL), start=time(NULL);
```

Рис. 3: Содержимое файла реализации сервера, начало.


```
clock_t now=time(NULL), start=time(NULL);
while(now-start<30)
{
    while((n = read(readfd, buff, MAX_BUFF)) > 0)
    {
        if(write(1, buff, n) != n)
        {
            fprintf(stderr, "%s: Ошибка вывода (%s)\n", __FILE__, strerror(errno));
            exit(-3);
        }
    }
    now=time(NULL);
}
printf("server timeout, %li - seconds passed\n", (now-start));
close(readfd); /* закроем FIFO */

/* удалим FIFO из системы */
if(unlink(FIFO_NAME) < 0)
{
    fprintf(stderr, "%s: Невозможно удалить FIFO (%s)\n", __FILE__, strerror(errno));
    exit(-4);
}

exit(0);
}
```

52,1 Внизу

Рис. 4: Содержимое файла реализации сервера, конец.

```
#include "common.h"

#define MESSAGE "Hello Server!!!\n"

int main()
{
    int msg, len, i;
    long int t;

    for(i=0; i<20;i++)
    {
        sleep(3);
        t = time(NULL);
        /* Баннер */
        printf("FIFO Client...\n");

        /* получим доступ к FIFO */
        if((msg = open(FIFO_NAME, O_WRONLY)) < 0)
        {
            fprintf(stderr, "%s: Невозможно открыть FIFO (%s)\n",
                __FILE__, strerror(errno));
            exit(-1);
        }

        /* передадим сообщение серверу */
        len = strlen(MESSAGE);
        if(write(msg, MESSAGE, len) != len)
        {
            fprintf(stderr, "%s: Ошибка записи в FIFO (%s)\n",
                __FILE__, strerror(errno));
            exit(-2);
        }

        /* закроем доступ к FIFO */
        close(msg);
    }
    exit(0);
}
```

Рис. 5: Содержимое файла реализации клиента 1.

```
#include "common.h"

#define MESSAGE "Hello Server!!!\n"

int main()
{
    int writefd; /* дескриптор для записи в FIFO */
    int msglen;
    long long int t;

    for(count=0; count<5;++count)
    {
        sleep(5);
        t = (long long int) time(0);
        /* баннер */
        printf("FIFO client...\n");

        /* получим доступ к FIFO */
        if((writefd = open(FIFO_NAME, O_WRONLY)) < 0)
        {
            fprintf(stderr, "%s: Невозможно открыть FIFO (%s)\n",
                __FILE__, strerror(errno));
            exit(-1);
        }

        /* передадим сообщение серверу */
        msglen = strlen(MESSAGE);
        if(write(writefd, MESSAGE, msglen) != msglen)
        {
            fprintf(stderr, "%s: Ошибка записи в FIFO (%s)\n",
                __FILE__, strerror(errno));
            exit(-2);
        }

        /* закроем доступ к FIFO */
    }
    close(writefd);
    exit(0);
}
```

Рис. 6: Содержимое файла реализации клиента 2.

```
all: server client

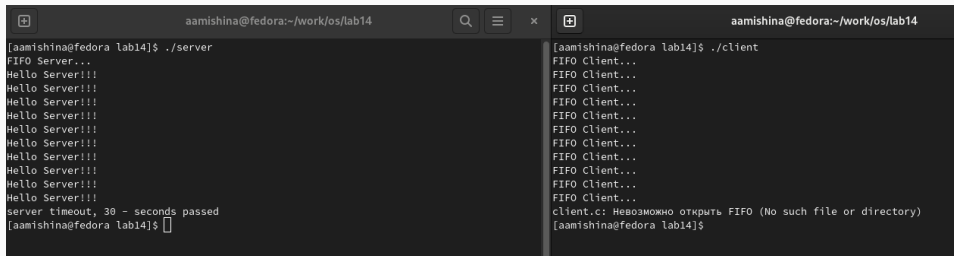
server: server.c common.h
    gcc server.c -o server

client: client.c common.h
    gcc client.c -o client

clean:
    -rm server client *.o
```

Рис. 7: Содержание Makefile.

Запуск программы



The image shows two terminal windows side-by-side. The left window is titled 'aamishina@fedora: ~/work/os/lab14' and shows the execution of a server program. The right window is also titled 'aamishina@fedora: ~/work/os/lab14' and shows the execution of a client program.

```
[aamishina@fedora lab14]$ ./server
FIFO Server...
Hello Server!!!
Hello Server!!!
Hello Server!!!
Hello Server!!!
Hello Server!!!
Hello Server!!!
Hello Server!!!
Hello Server!!!
Hello Server!!!
Hello Server!!!
server timeout, 30 - seconds passed
[aamishina@fedora lab14]$
```

```
[aamishina@fedora lab14]$ ./client
FIFO Client...
FIFO Client...
FIFO Client...
FIFO Client...
FIFO Client...
FIFO Client...
FIFO Client...
FIFO Client...
FIFO Client...
FIFO Client...
FIFO Client...
client.c: Невозможно открыть FIFO (No such file or directory)
[aamishina@fedora lab14]$
```

Рис. 8: Запуск программ на двух консолях.

- В ходе выполнения лабораторной работы я приобрела практические навыки в работы с именованными каналами.